Mimicking Inferior Temporal Representations in Deep Convolutional Neural Networks

Quintin Hansen
Department of Psychology
qhansen@ualberta.ca

1 Introduction

Deep convolutional neural networks (DCNNs), who's architecture is directly inspired by the mammalian visual system (MVS), have become the state of the art for object recognition tasks in the last decade. Despite this rapid advancement, DCNNs have some critical shortcomings, notably the types of error's made by DCNNs can differ dramatically from those made by human's [1]. Which leads to an important question regarding the similarity between the way visual information is processed in DCNNs compared to that of the MVS. A study by Razavi et al. [2] addresses this inquiry and has shown that the representations learned in DCNNs are similar to those found in the MVS. They also found that the representational similarity to the MVS increases as DCNNs improve in object recognition performance.

Given these findings one may wonder if directly training a DCNN to be more brain-like would boost it's performance. Federer et al. [3] sought out to answer this question and their results showed that training a DCNN to increase it's representational similarity to that of the neural activity in V1 of a monkey's visual cortex improved it's performance in an object recognition task. Here they used cosine similarity as the similarity metric and they used a DCNN where the architecture more closely imitates that of the MVS. Another study that extended the work by Federer et al.'s used a method involving correlations instead of cosine similarity as the metric. However, the correlation based method was not as successful in improving the DCNNs performance. Given the success of using neural data recorded from V1 of a monkey to improve performance of DCNNs, would the use of human fmri data improve on this further? This work seeks an answer by exploring the use of human fmri data from ventral regions of the temporal cortex to improve DCNN performance.

1.1 Neuroanatomy of Object Recognition

The current accepted model of object recognition in the MVS involves the ventral pathway, where visual information is hierarchically processed first in V1, then V2, V4 and finally the inferior temporal cortex (IT). In reality this is an oversimplified view since this pathway involves many recurrent connections and connections which bypass intermediate areas, however the hierarchical structure of this pathway is the focus for this work. Neurons in V1 have been shown to respond selectively to simple features such as edges in specific orientations [4]. V1 sends many feedforward projections to V2 for further processing where cells encode similar features to V1 as well as more complex features. Cells in V4, which receive feedforward projections from

V2, encode features of intermediate complexity such as shapes but not complex features such as objects and faces which are instead processed in the IT.

Haxby et al. [5] showed that neural activity in ventral temporal regions measured by fmri during an image viewing task was more correlated with activity measured while the participant viewed images of the same category than those for different categories. Which emphasizes the role neural activity in IT plays in encoding categorical information. Further, the study by Razavi et al. I mentioned earlier used DCNNs to specifically model IT representations, and it was the representational similarity to IT that correlated positively with their highest performing models. Given these pieces of evidence which highlight the importance of IT in object recognition and the ability for DCNNs to learn similar representations, I hypothesize that training a DCNN to learn representations more similar to that of the human IT would increase object recognition performance.

1.2 Previous Work: Representational Similarity Matrices (RSMs)

In the work done by Federer et al. they used a composite cost function to train the CORNet-Z model to both decrease it's classification cost and increase it's representational similarity to the neural data. The neural data comes from direct cell recordings from V1 of a monkey's visual cortex while the monkey viewed different images. To compare the representations for the latter part of the cost function they first computed the RSM from the neural data where for each pair of images *i* and *j* the cosine similarity between the neural data v_i and v_j is calculated using the formula below:

$$RSM_{i,j} = \frac{v_i \cdot v_j}{||v_i|| \times ||v_j||} \tag{1}$$

During training, for the similarity portion of the composite cost function a batch of images from the same set of shown to the monkey is input into the network to produce the hidden representations. For each image *i* and *j* that is compared in the batch the cosine similarity between the hidden representations h_i and h_j is computed which forms the RSM for the neural data denoted as $\widehat{RSM}$. The loss for the similarity portion of the composite cost function is the sum of squared errors between the cosine similarity from the neural data and from the hidden representations. This is to train the model to learn brain-like representations. The similarity portion is discounted with a dynamically updated λ value and summed with the classification cost to produce the composite cost function below:

$$cost = \lambda \sum_{i,j} (RSM_{i,j} - \widehat{RSM_{i,j}})^2 + \sum_{i_c} \hat{y}_{i_c} \log(y_{i_c}) \tag{2}$$

The classification cost represented by the latter portion of the composite cost is minimizes the cross-entropy between the network's predictions and the true output labels. The input images for the classification portion are from a different set of images than those for the RSM portion which is why I differentiated i (images for RSMs) and i_c. Instead i_c represents images from the CIFAR100 dataset which contains 100 classes of images. Federer et al. found that dynamically updating λ throughout training to keep the ratio parameter r constant produced the best results, where r is defined as:

$$r = \lambda \Big[\sum_{i,j} (RSM_{i,j} - \widehat{RSM_{i,j}})^2 \Big] \Big/ \Big[\sum_{i_c} \hat{y}_{i_c} \log(y_{i_c}) \Big] \tag{3}$$

The training procedure used by Federer et al. involved only applying the composite cost function for the first 10 epochs of training, then applying only the classification portion for the rest of the total 100 epochs to reduce computation. They tried a variety of r values between 0.01 and 4 and found that $r = 0.1$ performed best, outperforming the models trained with no neural data. They also tried applying the similarity cost to layers of the CORNet-Z model other than V1 but found that this performed worse than a model trained with no neural data.

1.3 Previous Work: Correlation Instead of Cosine Similarity

The work done by Sahir follow's that of Federer et al., using the same monkey V1 neural data; however, with two major differences. First, since the neural data was collected from 10 different sessions PCA was applied to the data for each session to produce the first 16 principle components in an attempt to align the the dimensions. Second, instead of computing RSMs to compare similarities Sahir's method computed the correlation between the neural representations and the hidden representations directly. To compute the correlation portion of the cost function the procedure goes as follows:

1. Apply PCA to neural data for that corresponds to each image i to get a vector v_i of 16 principle components $v_i = PCA(n_i)$
2. Input the same image i through the CORNet-Z model to get the hidden representation h_i from the model's V1 block
3. h_i and v_i are computed for each image and a set of weights W_v, W_h are learned on each step of batch data such that it maximizes the correlation:
$$(W_v, W_h) = \arg\max_{W_v, W_h} corr(W_v^T V, W_h^T H) \tag{4}$$
4. Using these weights the total composite cost function is as follows:
$$cost = \lambda \sum_i -Corr(w_v v_i, w_h h_i) + \sum_{i_c} \hat{y}_{i_c} \log(y_{i_c}) \tag{5}$$

Again I used the i_c term to differentiate images presented to the monkey and the CIFAR100 images used for the classification task. The λ value is computed using a constant r value in the same fashion as in Federer et al.'s work except using the negative correlation term instead of the RSM error term.

This method was attempted with differing *r* values from 0.001 to 1; however, all of the values failed to outperform the model trained without neural data with an *r* value of 0.001 coming the closest but still slightly worse. Although the results of this method weren't promising it, the idea is intriguing and it was still used in my work to attempt differing metrics for applying human fmri data to a DCNN.

2. Methods

2.1 Human Ventral Temporal Cortex FMRI Data

The FMRI data I used comes from the study by Haxby et al. I mentioned in section 1.1 which was collected from 6 human subjects viewing pictures from 7 different categories including faces, cats, and 5 different categories of man made objects. The region of interest (ROI) was a subset of voxels located in the ventral temporal cortex where the activity in this region correlates to categorical information processing of visual stimuli. The ROI included voxels from areas such as the fusiform face area and inferior temporal gyri. Each subject underwent 12 different sessions where they viewed one image from each of the 7 different categories. Each of the sessions produces 7 blocks of 10 time steps where the subject is viewing a single image during the 10 time steps and the subject is viewing a different image during each of the blocks. So In total each subject viewed 84 different images, additionally the same set of 84 images were displayed to each subject.

For each of the subjects the data was first centred by subtracting the mean values of the BOLD signal across all trials by voxel. The rest scans were removed then for each of the subject's 84 stimulus blocks (where each block is shaped #-voxels by 10 time steps) the mean was taken across time leaving 84 vectors for each subject labeled by image. The rest of the pre-processing depends on the similarity metric I used and is explained further below.

2.2 First Method: RSMs Using FMRI Data

For each subject an RSM was constructed by taking the cosine similarity between the each pair of FMRI vectors v_i and v_j which correspond to images *i* and *j* respectively (as seen in (1)). This results in 6 RSMs (shaped 84x84) which are then averaged across subjects to produce a single mean 84x84 RSM.

I performed all my experiments using the CORNet-Z DCNN architecture which has 4 blocks denoted as V1, V2, V4, and IT followed by a 3 fully connected layers with ReLU activations for the first two a softmax output layer. Each of the blocks consist of a single convolutional layer with a ReLU activation, next local response normalization is applied, and finally max pooling.

The composite cost function is identical to that used in Federer et al.'s work as described in (2) and (3).

To train the network a static learning rate of 0.01 was used to train the models all of which were optimized using batch gradient descent. I used a batch size of 128 for the batches of labeled CIFAR100 images and a batch size of 50 for the batches of image pairs and their corresponding cosine similarities. Each of the 3 fully connected layers also had dropout regularization applied with a 50% retention rate to prevent overfitting. This configuration of hyper-parameters is the same configuration described and used by Federer et al.. I attempted different combinations of learning rate and dropout retention rates but the configuration above seemed to give the best results.

Just as described by Federer et al. all networks were trained using the composite cost function for the first 10 epochs, while the remaining epochs only had the classification cost applied. I also tried different configurations of which epochs to start applying the composite cost and stop applying the composite cost, however I found no advantage over starting at epoch 0 and stopping at epoch 10. The total number of epochs the networks were trained for was a maximum of 100, some of the experiments were stopped early (~60 epochs) when overfitting was evident to save time and compute resources.

Since the ROI for the FMRI data was the ventral temporal cortex, the main focus of experimentation was to apply the similarity cost function to the IT block of the DCNN; however, I also experimented with applying it to the V1 and V4 blocks. Additionally, for direct comparison with the results from Federer et al. I also applied their monkey V1 data to the V1 block of the DCNN. For each of these configurations various *r* values were tested and the models were trained 6 times from different random initial conditions to ensure the results are not dependent on the weight initialization.

2.3 Second Method: Correlations Using FMRI

The FMRI data was first pre-processed as described in section 2.1 which resulted in 84 vectors with of fmri data for each subject. Next these vectors were stacked to form a 2D array shaped 84 by #-voxels for each subject. Since PCA is sensitive to scale, each array was scaled such that the mean was 0 and the standard deviation was 1. Then PCA was applied for dimensionality reduction using the first 16 principle components which resulted in a 16 by 84 array for each subject. The basic steps taken to make these array's were very similar to the steps used by Sahir to preprocess the monkey V1 data.

I used the same CORNet-Z DCNN architecture used by Federer et al. and Sahir which I described in section 2.2. Additionally the same hyper-parameters were used in these experiments. The cost function used for these experiments mimicked that of Sahir's work as seen in (4) and (5). Similar to the RSM experiments the main focus for these experiments was also to apply the correlation cost to the IT block of the DCNN, but I also attempted to on V4 and V1. Due to the poor performance of the monkey V1 data in Sahir's work, and the fact that I have the results obtained by Sahir, I did not see any value in running experiments using the monkey data (with this method) for comparison with the FMRI data. For each of the blocks of the DCNN combined with a variety of *r* values I trained 6 models for each *r* value such that the results are not dependent on the weight initializations.

3. Results

In addition to the 100 class labels for the images in the CIFAR100 dataset which are used for the cost function, the images are also labeled with one of 20 superclasses. For example: images labeled "racoon" or "possum" will be in the superclass of "small mammals". The accuracy calculated using the class labels is referred to as the "fine" accuracy and the accuracy calculated with the superclass prediction is referred to as "coarse" accuracy. The coarse accuracy is useful for determining the type of error's the model is making; for instance, Federer et al.'s results showed that the models trained with neural data made more within-superclass errors compared to the model trained without neural data. In my experiments, the fine accuracy and coarse accuracy scores measured on using images from the test set (images the model hasn't seen) after each epoch of training. I then plotted the mean ± SEM fine and coarse accuracy scores for each *r* value.

3.1 Results from RSM Method

The FMRI data applied to the IT block of the DCNN was attempted with ratios including 0 (no neural data), 0.001, 0.01, 0.1, 0.5 and 1. In the no neural data condition, the models were only trained for ~50 epochs because past this point overfitting was prevalent and the test accuracy metrics consistently levelled off at this point. In the experiments with ratio's greater than 0 there was a lot of variability between trials so they were all trained for 100 epochs. The same conditions above were also applied to the V4 block.

Applying the FMRI data to the IT block did not increase performance over no neural data, of the *r* values that are greater than 0, 0.001 and 0.01 performed similarly well. With an *r* value of 0.1 the resulting accuracy curves were much more varied rendering the mean +/- SEM over 6 initializations uninformative. However, inspecting the individual learning curves reveals some peculiarities which may provide hints as to why the FMRI data is decreasing the models performance.

The accuracy curves of three of the six models trained on the IT block with a ratio of 0.1 didn't gain accuracy scores much greater than chance throughout the 100 epochs of training; however, the accuracy scores of the other three shown in figure 2 eventually begin to start increasing their accuracy at seemingly random periods. Even though the representational similarity cost isn't applied during gradient descent after epoch 10, it is still computed and logged during the rest of training. Comparing the plots of figure 2 A and B, It seems like the increase in accuracy

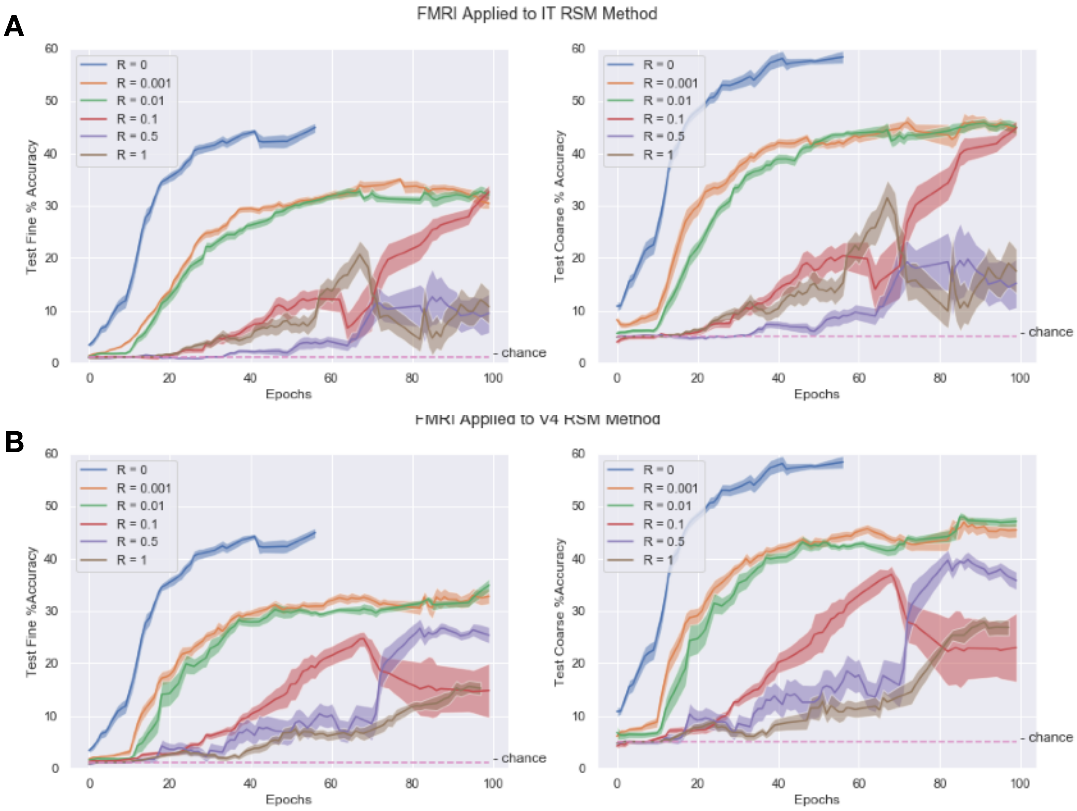

Figure 1: Fine classification accuracy (left) and coarse accuracy (right) measured on the test set for architectures trained with different weighting ratios, *r*, applied to human ventral temporal representational similarity for the first 10 epochs of training. Fine accuracy is the % of model's predictions in that are the correct class and coarse accuracy is the % of predictions that are the correct superclass where chance accuracies, indicated by the dotted line, are 1% and 5% respectively. The shaded areas are +/- SEM over 6 initializations. A) The representational similarity cost was applied to the IT block of the DCNNs. B) The representational similarity cost was applied to the V4 block of the DCNNs.

correlates with an increase in representational similarity cost; although, I'm not claiming that there's statistically significant relationship. When considering the plots in figure 1, the increase in accuracy above chance only occurs after the 10th epoch for those with ratios greater than 0. Together this may suggest that the FMRI representational similarity is often slowing down the ability for the network to train properly.

When the fmri data was applied to V4 similar variability to in results to IT occurred. However, the relationship between representational similarity cost and the accuracy didn't present a similar relationship to that shown in the IT experiments. I also did a few trials where FMRI data was also applied to V1 and this out performed V4 and IT and lacked the wild variations but it did not outperform models trained with no neural data.

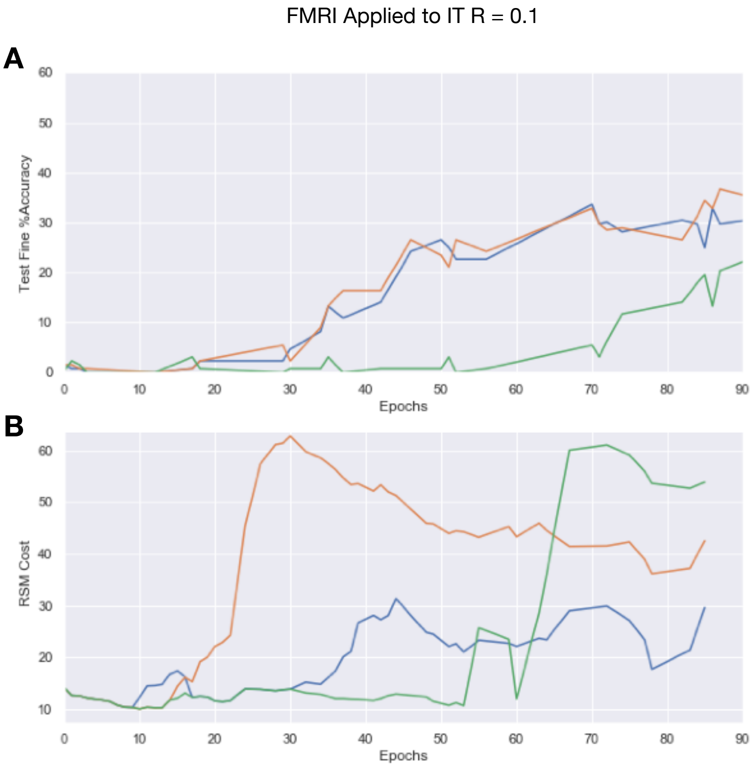

Figure 2: A) Fine classification accuracy plotted by epochs where the representational similarity cost is applied to IT with a ratio of 0.1. B) The representational cost plotted by epoch (A sliding window of 3 epochs where the actual plotted value is the mean was used to slightly smoothen the curve). The cost and the accuracy are matched by the colour of the curves. This is subset of 3 models from the 6 total that were trained which were chosen based on the highest accuracy scores at epoch 90.

3.2 Results from Correlation Method

Applying the human ventral temporal FMRI data to the V4 block of the DCNN, the results from all trials was chance accuracy. When applying the FMRI data to the IT block there was more variation in the results; however, the results were still much worse than when no neural data was applied.

I also tried applying the FMRI data to the V1 block using this method with ratios 0.001 0.01 and 0.1 which performed better than when it was applied to V4 or IT. However, the performance was significantly worse than the performance Sahir reported using the monkey V1 data.

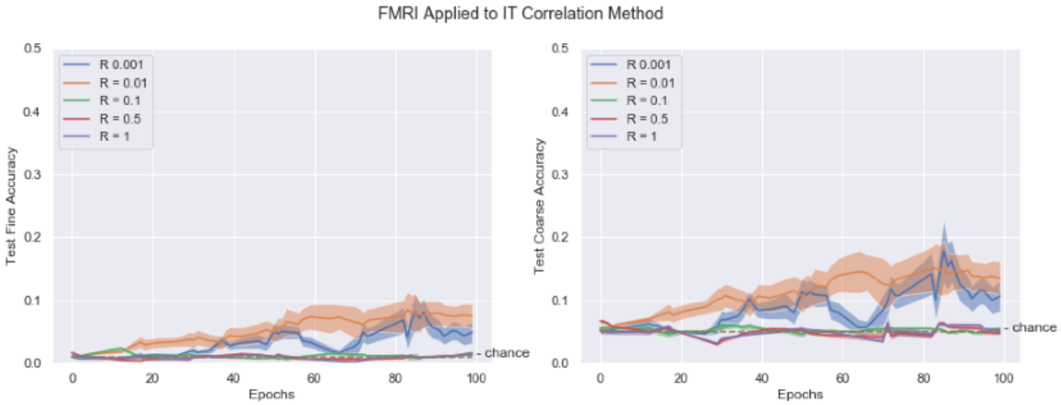

Figure 3: Fine classification accuracy (left) and coarse accuracy (right) measured on the test set for architectures trained with different weighting ratios, r, applied to human ventral temporal correlation cost to the IT block for the first 10 epochs of training.The shaded areas are +/-

4 Discussion

The results clearly show that the FMRI data applied to the either the V1, V4, or IT block of the DCNN with either of the similarity metrics described failed to increase object recognition performance. Nevertheless, I do believe that it is worth considering the possible reasons for the lack of performance enhancement. When Razavi et al. demonstrated that the representational similarity of DCNN models most closely resembled human IT, they didn't use cosine similarity or distance for their representational similarity analysis [5]. They instead the correlation distance. So one approach which may be promising for future experiments using correlation distance instead of cosine similarity/distance to produce the RSMs and then use the same composite cost function from Federer et al.'s work.

Bobadilla-Suarez et al. [6] investigated the differences between performance of various similarity metrics in representing FMRI data in comparison to the standard similarity metric for this type of data, namely the Pearson correlation. They found that Mahalanobis distance and Minkowski distance both out performed Pearson correlation and cosine distance for representing similarity between brain states. The method they used to investigate the performance of the different metrics is very intriguing and if it's shown to be valid it could prove useful in discovering the best similarity metrics for this specific application.

Looking closer at the correlation cost when using Sahir's method used on the FMRI data I noticed that the cost often quickly moved dropped to -50 and continued at this level past epoch

10 (where the correlation cost seizes to contribute to optimization) and continued at this level till training stopped at epoch 100. The classification performance for the many experiments that showed this behaviour (which made up more than 50% these trials) also never increased past chance accuracy. Since the batch size for the correlation portion of the cost function was 50, this means that the correlation between value for each $Corr(w_v v_i, w_h h_i)$ term in the batch was 1. Indicating that the optimal weights W_v, W_h for maximizing correlation was found for each subsequent step of training. If the maximum correlation is already achieved before applying gradient decent to improve the cost via changing the networks weights and thus the affecting the hidden representations then the this portion of the cost is rendered ineffective. This should be investigated further to determine if it's a bug in the code or whether it's a fundamental flaw of this method.

Despite the lack of positive results in using the human ventral temporal FMRI data to improve DCNN performance, this work illuminates many avenues for future research that may be more promising. Considering the many short comings our best DCNNs have such as differences in patterns of error's made by these models when compared to error's humans make, or the vulnerability to adversarial examples, having a model that processes visual information similarly to humans is highly desirable. Some have argued that differences between visual processing in DCNNs and that of humans is advantageous evidenced by their superior performance on some tasks such as medical imaging diagnosis. And I agree that this can be an advantage in some situations, but in applications such as autonomous vehicles when the model makes mistakes a human wouldn't have it's very unlikely to gain trust from the public. With the many potential practical applications as motivation, more research is needed to thoroughly explore the implementing human-like processing in artificial intelligence systems.

References

[1] Pramod, R. & Arun, S. Do computational models differ systematically from human object perception

[2] Khaligh-Razavi, S. M., & Kriegeskorte, N. (2014). Deep supervised, but not unsupervised, models may explain IT cortical representation. *PLoS computational biology*, *10*(11), e1003915. https://doi.org/10.1371/journal.pcbi.1003915

[3] Federer, C., Xu, H., Fyshe, A., & Zylberberg, J. (2020). Improved object recognition using neural networks trained to mimic the brain's statistical properties. *Neural Networks*, *131*, 103-114.

[4] Hubel, D. H., & Wiesel, T. N. (1968). Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, *195*(1), 215–243. https://doi.org/10.1113/jphysiol.1968.sp008455

[5] Haxby, J., Gobbini, M., Furey, M., Ishai, A., Schouten, J., and Pietrini, P. (2001). Distributed and overlapping representations of faces and objects in ventral temporal cortex. *Science 293*, 2425–2430.

[6] Bobadilla-Suarez, S., Ahlheim, C., Mehrotra, A. *et al.* Measures of Neural Similarity. *Comput Brain Behav 3***,** 369–383 (2020). https://doi.org/10.1007/s42113-019-00068-5